

~~132~~
~~02~~ Cont'd

REMARKS

I. Information Disclosure Statement

Page 8 of 22
Ehnebuske et al. – 09/204,971

II. Specification

The Office Action objects to the specification for various reasons. By this Response, the Related Applications paragraph is amended to include the serial numbers of the co-pending patent applications. In addition, the title of the application is amended to be more descriptive.

Applicants however, disagree with the Examiner's apparent objection to the use of various statements in the specification. Specifically, the Examiner objects to the statement "Typically, these rules always have to be true except possibly during some specific rule free period such as during the middle of a business operation" on page 6 as not being supported by the specification. The Examiner's objection is unclear. It is not understood how a statement that is in the specification is allegedly not supported by the specification. However, in an effort to clarify this statement for the Examiner, Applicants offer that this statement means that normally, for integrity to be maintained in a system, the integrity rules must all have an outcome that is "true." The only time when this requirement may be relaxed is possibly during a "rule free" period. An example of a "rule free" period is during the middle of a business operation. The statement is supported by the specification. In fact, the statement supports itself.

The Examiner also objects to the use of the term "point of variability" on page 10 alleging that this term is not clearly defined. Applicants respectfully disagree. The next statement on page 10 clearly describes the point of variability as a point in an enterprise operation at which rules may be associated to implemented variable behavior. This may be done, for example, by associating different rules with the point of variability, by changing the process for rules already associated with the point, or the like. The specification is clear as to what this term means and the Examiner's allegation that the term is not clearly defined is without merit.

The Examiner also objects to the use of the term "participant" at page 8 of the specification as not being a term of art. The term "participant" as it is used in the present application is clearly defined on page 11, lines 25-27 as being elements of a unit of work that are modified by the processing carried out in association with the unit of work. Thus, the term "participant" is not unclear.

The Examiner also objects to the use of the statement “Natural structuring is provide whether or not each unit of work is totally or partially automated,” on page 11 although the Office Action does not state why the Examiner objects to this statement. However, for the Examiner’s understanding, as stated two sentences prior to the one in question, a unit of work may follow the structures of work done by employees and associates of businesses. The statement in question merely states that natural structuring, in other words structuring as if the work were done by a employees and associates of businesses, is used regardless of whether the unit of work is totally or partially automated. Therefore, this statement is not unclear.

The Examiner further objects to the term “disappear” on page 12 as not being a “normal term in the art.” The term “disappear” as it is used in the context of page 12 means, as the Examiner recognized, that the changes are not kept. Thus, the term is not unclear.

Lastly, the Examiner objects to the statement “visible to all subsequently started unit of work” on page 12. The objection to this statement reads “Scope rules as per above” which does not make any sense. Applicants do not understand what the Examiner is attempting to say here. However, in an effort to clarify the statement objected to by the Examiner, this statement means that all subsequent units of work will be made aware of the change in state. Thus, this statement is not unclear.

In view of the above, Applicants respectfully submit that the specification is not unclear and that the objections should be withdrawn. Accordingly, Applicants respectfully request withdrawal of the objection to the specification.

III. Rejection under 35 U.S.C. § 112, Second Paragraph

The Office Action states that the claims are rejected under 35 U.S.C. § 112 but fails to make any note of which claims are being rejected or the basis upon which they are being rejected. Thus, the Examiner has not satisfied the burden of establishing a rejection under 35 U.S.C. § 112. As a result, Applicants are under no obligation to provide any amendments or arguments to overcome this alleged rejection.

In this section of the Office Action, the Examiner sets forth a number of different “interpretations” of the terms used in the claims. The terms in the claims must be interpreted in light of the specification as one of ordinary skill in the art would interpret these terms. Thus, the Examiner’s statements in this section with regard to his personal interpretation are not limiting to the present claims. However, in an effort to clarify the use of these terms to aid the Examiner, Applicants offer the following example definitions. These definitions are not meant to imply any limitations to the use of these terms in the claim but are only offered as example definitions obtained from the current specification.

“Checking State Integrity” – checking to see if the states of the participants in the unit of work maintain some feature that is important to the integrity of the application state and based on the collective judgment of the rules, determining if the unit of work as a whole passes the integrity check (page 12, lines 17-23);

“Externalized Rules” – *****

“Unit of Work” – representations of pieces of business work which define each business context in which they are carried out (page 11, lines 21-23; also an example is shown in Figure 3);

“Commit” – a process of determining whether to keep the changes in state made by the processing of a unit of work (page 12, lines 11-14; page 13, lines 19-23); also to keep the changes in state by storing them in persistent storage (page 12, lines 25-28);

“Abort” – to stop processing or to cause a process to not complete (page 12, lines 1-3);

“Participant” – see the definition of participant set forth above in the response to the objections to the specification;

“Disappear” – see the definition of disappear set forth above in the response to the objections to the specification;

“Visible to all subsequently started unit of work” – see the definition of this phrase set forth above in the response to the objections to the specification;

“State of the enterprise application” – the state of the enterprise application as stated by the Examiner on page 6 of the Office Action.

In this section of the Office Action, the Examiner also provides some allegedly “common knowledge of programming” terms and definitions obtained from the Martin reference and the IBM Computer Dictionary. It is not clear whether the Examiner is asserting a rejection of the claims or not in this section. If the Examiner is stating a rejection, the Examiner is required to set forth which claims are being rejected and the basis for each rejection. Therefore, Applicants need not respond to the Examiner’s statement of supposed “common knowledge of programming” terms.

In an order to expedite matters, however, Applicants do not disagree that these definitions are possible definitions of the terms. Regardless, the terms as used in the pending claims are defined by the specification regardless of the definitions chosen by the Examiner. Thus, Applicants’ claims are not limited to the definitions obtained from the Martin reference or the IBM Computer Dictionary.

In this section of the Office Action, the Examiner also provides a subsection entitled “Terms of the Applicant’s vs. Martin Reference.” It is not clear what the Examiner is attempting to state in this section other than the Martin reference allegedly teaches “Integrity Rules.” The Examiner has not stated how this alleged teaching affects the present application. Because the Examiner has not set forth a clear rejection in this section of the Office Action, Applicants are under no obligation to respond. However, it should be noted that Applicants are not bound by any statement set forth in this section of the Office Action and respectfully traverse any allegation with regard to any interpretation by the Examiner of terms or phrases used in the present application that are not supported by the present specification. Again, the present application and the pending claims are not limited to any definitions found in the Martin reference.

IV. 35 U.S.C. § 103, Obviousness

The Office Action rejects claims 1-29 under 35 U.S.C. § 103(a) over Martin, Principles of Object-Oriented Analysis and Design, 1993 in view of alleged common knowledge of programming as taught by Complete Code and definitions as provided by the IBM Computer Dictionary. This rejection is respectfully traversed.

With regard to claim 1, the Office Action states:

Martin teaches a method for performing general integrity checks using rules in an application running on a data processing system (**Martin**, page 133, 138 – 142, structure of Rules) comprising: identifying a point in a unit of work where application state integrity is to be verified (**Martin**, page 143, object state rules), wherein the unit of work includes a plurality of participants (**Martin**, page 143, last paragraph, linked to appropriate items in OO diagram); obtaining rules associated with each participant in the unit of work (**Martin**, page 144, Box 10.3); responsive to obtaining the rules (**Martin**, page 136, Rules Linked to Diagrams), running the rule obtained for each of the participants to verify the integrity of an application state (**Martin**, page 143, object state rules), according to the plurality of participants; general integrity checks running on a data (**Martin**, page 133, 138 – 142, structure of Rules). **Martin** does not explicitly teach the programming constructs of responsive to determining that the unit of work is to be completed. Although, **Martin** clearly supports Rules and the testing for conditions the reference does not explicitly states RULES can be used to determine if a write operation should be performed or not. The following overviews the teaching of **Martin**.

Martin Reference teaches RULES

Testing the Integrity of an Object (page 143)

When condition (Chapter 10 a format for Rules logic structure testing the state of the Object)

Perform operation

When condition

Perform operation

(**Note**: the construct closely resembles the structure of a CASE/switch statement in many languages)

What **Martin** does give is the endless possibilities of the conditions and the endless possibilities of the operations, such as if a state condition is wrong don't write (negative results) if the state operation is correct (positive results) perform a write operation. Examiner, holds determinations when to perform a write operation and when not to perform a write are issues of normal use and considered part of being a artisan of ordinary skill in the art. The term for employing common sense in programming is called Defensive Programming. It is the reference "**Code Complete**" by Steve McConnell that teaches defensive programming like on page 97 of **Code Complete** "Garbage In Does Not

Mean Garbage Out”. In other words test before you use data. Therefore, it would have been obvious to one of ordinary skill at the time of invention to use the teaching of Martin’s RULES to perform defensive programming, because “...it’s the recognition that programs will have problems and modifications, and that a smart programmer will develop code accordingly”. (**Code Complete**, page 94, 5.6 Defensive Programming, Key Point).

Claim 1, which is representative of claims 19 and 25 with regard to similarly recited subject matter, reads as follows:

1. A method for performing general integrity checks using rules in an application running on a data processing system comprising:
 - identifying a point in a unit of work where application state integrity is to be verified, wherein the unit of work includes a plurality of participants;
 - responsive to determining that the unit of work is to be completed, obtaining rules associated with each participant in the unit of work; and
 - responsive to obtaining the rules, running the rules obtained for each of the participants to verify the integrity of an application state, according to the plurality of participants.

It should first be noted that the Martin reference is a reference comprising over 400 pages of text. Therefore, Applicants in this Response are only addressing those sections explicitly cited by the Office Action.

Martin is a general text book that includes a section on the use of rules in object oriented programming. While Martin generally teaches the use of rules, and even integrity rules, Martin does not teach or even suggest the specific use of rules set forth in claim 1. A general teaching does not render every specific use of the general teaching obvious. The following are Applicants’ responses to the Office Action’s allegations regarding the specific features recited in claim 1. Martin does not teach or suggest any of these features.

The Office Action alleges that Martin teaches a method of performing general integrity checks using an application running on a data processing system at pages 133 and 138-142. While it is true that Martin does teach the use of integrity rules and states that these rules indicate that something must be true (page 137), Martin does not teach the

specific use of integrity rules set forth in claims 1, 19 and 25 (hereafter only referred to as claim 1), as discussed below.

The Office Action alleges that Martin teaches identifying a point in a unit of work where application state integrity is to be verified merely because Martin allegedly teaches object state rules at page 143. First, Martin does not teach or even suggest units of work as the term is used in the present application. As set forth above, a unit of work is a piece of business work which defines each business context in which it is carried out. While Martin teaches that business policies may be represented as rules (see page 133), Martin makes not mention or even suggestion regarding units of work.

Because Martin does not teach units of work, Martin cannot be found to teach or even suggest identifying a point in a unit of work where application state integrity is to be verified. While Martin teaches object state rules on page 143, all that is stated is “Object state rules are identified in object-structure analysis. They are associated with diagrams, such as the data-structure diagram, object-relationship diagram, or composed-of diagram.” It is not seen how such a general statement somehow teaches the very specific feature of identifying a point in a unit of work where application state integrity is to be verified, as recited in claim 1. The description of object state rules on page 143 of Martin does not make any mention of units of work, let alone identifying a point in a unit of work where state integrity is to be verified.

Furthermore, even if it were interpreted that Martin teaches units of work, Martin still does not teach identifying a point in a unit of work where application state integrity is to be verified. As noted above, the general statement that there are object state rules, that they are identified in object-structure analysis, and that they may be associated with diagrams, has nothing to do with identifying a point in a unit of work where application state integrity is to be verified. The Examiner is reading features into the Martin reference that is simply not there in order to arrive at Applicants’ claimed invention having first had benefit of Applicants’ disclosure. In other words, the Examiner is engaged in hindsight reconstruction in which the Examiner is conjuring teachings from the reference that simply are not there.

The Office Action alleges that Martin teaches that a unit of work includes a plurality of participants merely because Martin allegedly mentions “three more rule

windows that are linked to appropriate items in OO diagrams” (page 143, last paragraph). It is not understood how linking rule windows to object oriented diagrams teaches a unit of work having a plurality of participants. There is no correlation between the feature recited in claim 1 and the cited portion of Martin. Linking rule windows to object oriented diagrams has nothing to do with a unit of work having a plurality of participants.

The Office Action further alleges that Martin teaches obtaining rules for each of the participants in the unit of work merely because Martin provides example of rules associated with diagrams of object oriented analysis in Box 10.3 on page 144. Box 10.3 of Martin does not teach a unit of work, let alone obtaining rules for each participant in the unit of work. The only thing that Box 10.3 teaches is examples of rules for different types of object oriented diagrams. There is no teaching or even suggestion in Box 10.3 of using units of work, and definitely no teaching or suggestion of obtaining rules for each participant in a unit of work. Again, the Examiner is engaging in hindsight reconstruction by reading in teachings to the Martin reference that simply are not there.

In addition, the Office Action alleges that Martin teaches running the rules obtained for each of the participants to verify the integrity of an application state merely because Martin allegedly teaches object state rules on page 143. The general teaching of object state rules does not provide any teaching or suggestion regarding running rules obtained for participants in a unit of work to verify the integrity of an application state. The general teaching of object state rules makes no teaching or suggestion of units of work, rules for participants in a unit of work, obtaining and running rules for participants in a unit of work, or that there is any correlation between rules for participants in a unit of work and an application state. There simply is no correspondence between the section of Martin cited by the Office Action and the features recited in claim 1.

In summary, Martin has nothing to do with the invention recited in claim 1. While Martin provides a number of general teachings, they do not teach or suggest any of the features recited in claim 1. Furthermore, the Examiner appears to be stretching the interpretation of Martin in an attempt to reach Applicants’ claimed invention beyond any reasonable interpretation of the reference. Such a stretch of interpretation is based on hindsight reconstruction using Applicants’ own disclosure as a guide.

Applicants do agree with the Office Action that Martin does not teach obtaining rules for each of the participants in a unit of work in response to determining that a unit of work is to be completed, as recited in claim 1. This is because Martin does not teach a unit of work, participants in a unit of work, or even that a unit of work may be completed. However, Applicants disagree with the Office Action regarding the allegation that Martin provides “endless possibilities” of conditions and operations such as write operations and that when to perform a write operation and when not to perform a write operation somehow makes obvious this feature. The Office Action also cites Code Complete as teaching defensive programming which the Office Action somehow links with Martin to allegedly modify the Martin reference so that a determination that a unit of work is to be completed is somehow made obvious.

First, the Applicant cannot follow the Examiner’s reasoning. It is not at all clear how a general teaching of conditions can be found to teach the specific feature set forth in claim 1 of a determination that a unit of work is to be completed. The Office Action states that the conditions of when to write and when not to write make this feature obvious. Applicants do not claim to be the first to invent determining if a unit of work is to be completed. Rather, Applicants claim obtaining rules for participants in the unit of work when it is determined that a unit of work is to be completed. Thus, the Office Action’s example of a write condition has no bearing on the presently claimed features.

Second, it is not clear how a general teaching of “defensive programming” may somehow be combined with the general teachings provided in Martin to arrive at the specific method set forth in claim 1. The Code Complete reference does not provide any teachings that cure the deficiencies in Martin set forth above. In other words, the general teaching of “defensive programming” does not render obvious “identifying a point in a unit of work where application state integrity is to be verified, wherein the unit of work includes a plurality of participants; responsive to determining that the unit of work is to be completed, obtaining rules associated with each participant in the unit of work; and responsive to obtaining the rules, running the rules obtained for each of the participants to verify the integrity of an application state, according to the plurality of participants,” as recited in claim 1.

Although not explicitly cited in the rejection of claim 1, the IBM Computer Dictionary provides no additional teaching regarding any of the features of claim 1. That is, the combination of the IBM Computer Dictionary with Martin and Code Complete does teach or suggest any of the features in claim 1. None of these references have anything to do with the presently claimed invention recited in claim 1. The Examiner has taken extremely general teachings in a text book and attempted to stretch these teachings beyond reasonable interpretation and read additional teachings into the references that are not there in an attempt to encompass the very specific method set forth in claim 1. Such a stretch of the teachings and reading in of additional teachings is not supported by the references and is based solely on hindsight reconstruction using Applicants' own disclosure as guide.

With regard to claims 8, 22 and 28, neither Martin, Code Complete, or the IBM Computer Dictionary teaches or suggests detecting a commit for a unit of work. As set forth above, a commit in this context is a process for determining whether to commit the state changes made by the unit of work to the participants in the unit of work. The Office Action alleges that the mere teaching of having conditional rules in Martin somehow is the same as detecting a commit for a unit of work. Applicants respectfully disagree. Yet again, the Examiner is reading in teachings to the Martin reference that are not there. The general teachings of conditional rules does not teach or suggest detecting a commit of a unit of work.

As previously noted above, Martin does not teach a unit of work. Therefore, Martin cannot teach detecting a commit of a unit of work. Furthermore, Martin does not teach a commit of a unit of work. A general teaching of a condition rule does not provide any teaching or suggestion of a determination of whether to commit the state changes made by a unit of work to the participants in the unit of work, i.e. a commit of a unit of work.

In addition, claims 8, 22 and 28 include features that are similar to some of the features in claims 1, 19 and 25. Therefore, Martin, Code Complete and the IBM Computer Dictionary do not provide any teaching or suggestion of these features in claims 8, 22 and 28, as discussed above.

With regard to claim 12, as previously noted, none of the references teach or suggest a unit of work, participants in a unit of work, or locating rules for the participants in response to activation of the unit of work to complete the unit of work.

In view of the above, Applicants respectfully submit that neither Martin, Code Complete, or the IBM Computer Dictionary, either alone or in combination, teach or suggest the features recited in claims 1, 8, 12, 19, 22, 25, 28. At least by virtue of their dependency on claims 1, 8, 12, 19, 22, 25 and 28, respectively, none of the references either alone or in combination teach or suggest the features set forth in dependent claims 2-7, 9-11, 13-18, 23-24, 26-27 and 29. Accordingly, Applicants respectfully request withdrawal of the rejection of claims 1-29 under 25 U.S.C. § 103(a).

In addition, none of the references teach or suggest any of the specific features set forth in the dependent claims 2-7, 9-11, 13-18, 23-24, 26-27 and 29. The Office Action provides specific rejections of each of the dependent claims based on the combination of references discussed above. Each of these rejections is based on the same flawed interpretation and reading in of teachings discussed above and therefore, are traversed for similar reasons as set forth above.

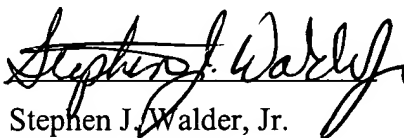
In addition, because the dependent claims build off of the features recited in their respective independent claims, the deficiencies of the cited references are likewise applicable to the additional features set forth in the dependent claims. For example, because none of the references teach or suggest a unit of work or running rules for participants in a unit of work, as recited in claim 1, none of the references can be found to teach “responsive to a negative result obtained by running the rules, aborting the unit of work,” as recited in claim 2.

III. Conclusion

In view of the above, Applicant respectfully submits that all of the claims are directed to allowable subject matter and that the application is in condition for allowance. The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

Respectfully submitted,

DATE: August 8, 2001



Stephen J. Walder, Jr.
Reg. No. 41,534
Carstens, Yee & Cahoon, LLP
P.O. Box 802334
Dallas, TX 75380
(972) 367-2001
Attorney for Applicant

Appendix of Claim Amendments
August 7, 2001

Please amend claims 1 and 7 as follows:

1. (Once Amended) A method for performing general integrity checks using rules in an application running on a data processing system comprising:

identifying a point in a unit of work where application state integrity is to be verified, wherein the unit of work includes a plurality of participants;

responsive to determining that the unit of work is to be completed, obtaining rules associated with each participant in the unit of work; and

responsive to obtaining the rules, running the rules obtained for each of the participants to verify the integrity of an application state, according to the plurality of participants[;].

7. (Once Amended) The method of claim 1, wherein at least zero integrity checking rules are associated with each participant within the plurality of participants[;].

Redacted Paragraph

August 7, 2001

Paragraph beginning on line 6 of page 1 that is replaced by the this Response reads as follows:

The present invention is related to applications entitled Method And Data Processing System For Specifying And Applying Rules To Classification-Based Decision Points In An Application System, attorney docket no. AT9-98-287, filed even date hereof, assigned to the same assignee; and Method And Apparatus For Applying Business Rules In An Object Model Driven Context, attorney docket no. AT9-98-266, filed even date hereof, assigned to the same assignee, which are incorporated herein by reference.